

Family Name: Other Names:

Student ID: Signature

COMP 112 : Test

2018, Apr 9 ** WITH SOLUTIONS **

Instructions

- Time allowed: **50 minutes**
- Attempt **all** the questions. There are 50 marks in total.
- Write your answers in this test paper and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- Brief Java documentation is provided with the test
- This test contributes 15% of your final grade
(But your mark will be increased to your exam mark if that is higher.)
- You may use dictionaries.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions

Marks

1. Writing programs with text input, output and if	[10]	<input type="text"/>
2. Writing programs with graphical output	[10]	<input type="text"/>
3. Writing methods that use objects	[10]	<input type="text"/>
4. Using String methods and if	[10]	<input type="text"/>
5. Using loops and lists	[10]	<input type="text"/>
	TOTAL:	<input type="text"/>

Question 1. Writing programs with text input, output and if**[10 marks]**

Complete the following calculateCost method to calculate the cost of renting a car.

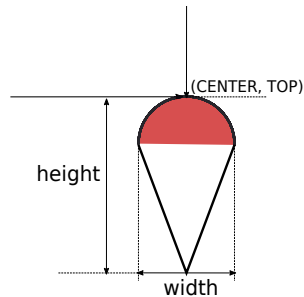
The method should

- ask the user for the number of days of renting a car
- calculate the cost based on the following rules
 - Renting a car costs \$50 per day.
 - if you rent the car for 7 or more days, you get 10% off your total.
 - if you rent the car for 3 or more days (but less than 7), you get \$10 off your total.
- print out the cost for the number of days in the form:
Rental cost for 4 days: \$190.00

```
public void calculateCost () {  
  
    int days = UI.askInt("How many days:");  
    double price = 50.0;  
    double cost = price * days;  
    if (days >= 7) {  
        cost = cost * (1-0.10);  
    }  
    else if (days >= 3) {  
        cost = cost - 10;  
    }  
    UI.printf("Rental cost for %d days is $%.2f.", days, cost);  
  
}
```

Question 2. Writing programs with graphical output**[10 marks]**

Complete the following drawIceCreamSign method to draw the following ice cream sign consisting of one arc (half circle) and two lines.



- The position of the sign is defined by the constants CENTER and TOP.
- The parameters of drawIceCreamSign specify the width and the height of the sign.
- Compute values for the left, right and bottom of the sign, and put them in variables. You may also use other variables to store other coordinates.
- Draw and fill the arc (half circle) and draw the two lines, using the variables.

```

public static final double CENTER = 400;
public static final double TOP = 50;
    :
public void drawIceCreamSign(double width, double height){

    double left = CENTER - width/2;
    double right = CENTER + width/2;
    double lineTop = TOP + width/2;
    double bottom = TOP + height;
    UI.fillArc ( left , TOP, width, width, 0, 180);
    UI.drawLine( left , lineTop, CENTER, bottom);
    UI.drawLine(CENTER, bottom, right, lineTop);

}

```

Question 3. Writing methods that use objects**[10 marks]**

This question is about a program that makes two bank accounts for two people and allows them to deposit or withdraw money.

The documentation for the Account class is given below.
It has one constructor and five methods:

```
// Constructor:
public Account(String name, int accountNo, double initialBalance )
/** Creates a new Account object of the specified name. e.g. "Tom",
    account number e.g. 1234, and an initial balance e.g. $100.0 */

//Methods:
public String getName()
/** returns the name of the account */

public double getBalance()
/** returns the balance of the account */

public void deposit(double amount)
/** makes a deposit : adds the specified amount to the balance */

public void withdraw(double amount)
/** makes a withdrawal: subtracts the specified amount from the balance */

public void printStatement()
/** prints the name, account number and balance of the account */
```

Complete the banking method on the facing page so that it

1. Creates two bank accounts.
 - Creates an account for Alex, with the account number 1349, and an initial balance of \$20.0 in his account.
 - Creates an account for Bob, with the account number 4987, and an initial balance of \$300.0 in his account.
2. Calls a sequence of methods on the two objects to complete the following actions:
 - Alex makes a deposit of \$100.
 - Bob makes a withdrawal of \$200.
 - Alex withdraws \$50 and gives it to Bob, and Bob put it in his account.
 - Print the statements for both Alex and Bob
 - Compare the balance of the two accounts, and print out the name of the person who has more money. Note: you need to call methods to get the balance or name of an account.

```
/** Creates two accounts then performs actions on the accounts */
public void banking(){
    Account a1 = new Account("Alex", 1349, 20.0);
    Account a2 = new Account("Bob", 4987, 300.0);
    a1.deposit(100);
    a2.withdraw(200);
    a1.withdraw(50);
    a2.deposit(50);
    a1.printStatement();
    a2.printStatement();
    if (a1.getBalance() > a2.getBalance()){
        UI.print(a1.getName() + " has more money in the bank");
    }
    else{
        UI.print(a2.getName() + " has more money in the bank");
    }
}
```

Question 4. Using String methods and if.**[10 marks]**

The `checkCode(String code)` method on the facing page should check whether its argument satisfies the following constraints and prints out

- “not valid”, if it fails any constraints or
- “valid”, if it satisfies all the constraints.

A string is a valid code if

- it has 7 characters, *and*
- the first 2 characters are either “EC” or “IM”, *and*
- the code does not include “:” or “;”
- the last 4 characters include the digit 1 or the digit 0

Your method will need to use methods from the String class on the last page of the documentation sheets, such as `length`, `substring`, `contains`, and `startsWith`.

(Question 4 continued)

```
public void checkCode(String code){
    int len = code.length();
    String part2 = code.substring(3);

    if ( ( len ==7 ) &&
        ( code.startsWith("EC") || code.startsWith("IM") ) &&
        !code.contains(":") && !code.contains(";") &&
        ( part2.contains("1") || part2.contains("0"))) {
        Ul.println ("Valid");
    }
    else {
        Ul.println ("Not Valid");
    }
}
```

Question 5. Using loops and lists**[10 marks]**

The question involves a method for “filling” a paragraph: given a list of words and a maximum line length, print out the words as a paragraph, separated by spaces, making sure that no line has more than the maximum number of characters in it (including spaces). Any words that are longer than the line length should start on a new line but should be broken by a hyphen.

For example, given a list containing the words:

"A", "way", "to", "fill", "a", "paragraph", "of", "extraordinary", "words"
if the maximum line length is 10, then the filled paragraph should be:

A_way_to
fill_a
paragraph
of
extraordi-
nary_words

(spaces shown as “_”)

(a) **[2 marks]** The fillTextBad method below attempts to fill a paragraph, but does not work correctly. Its parameters are the list of words and the maximum line length.

```
public void fillTextBad ( ArrayList<String> words, int max){
    int count = 0;
    for (String word : words){
        Ul. print (word + " ");
        count = count + word.length() + 1;
        if (count >= max){
            Ul. println ();    //This starts a new line
            count = 0;
        }
    }
    Ul. println ();
}
}
```

What will fillTextBad print out if given the list of words in the above example and a maximum line length of 10? Show the spaces using a “_”.

A_way_to_fill_
a_paragraph_
of_extraordinary_
words_

(Question 5 continued)

(b) [8 marks] Complete the fillText method below so that it works correctly.

```

public void fillText ( ArrayList<String> words, int max){
    int count = 0;
    for ( String word : words){
        while (word.length() >= max){ //word needs to be broken
            if (count>0){
                Ul. println ();
                count=0;
            }
            Ul. println (word.substring (0,max-1)+"-");
            word = word.substring(max-1);
        }
        if (count + word.length() >= max){ // need to start new line
            Ul. println ();
            count = 0;
        }
        if (count==0){ // no space if at beginning of line
            Ul. print (word);
            count = count + word.length();
        }
        else {
            Ul. print (" "+word);
            count = count + word.length() + 1;
        }
    }
    Ul. println ();
}
}

```

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.